# The Terascale Simulation Tools and Technologies Center

David Brown *(Lawrence Livermore)*

Lori Freitag Diachin *(Sandia)*

Jim Glimm *(Brookhaven, SUNY Stony Brook)*

*Presented at the ACTS Toolkit Meeting, August 2003*

*http://www.tstt-scidac.org/*

Terascale
Simulation
Tools
and
Technologies

# The TSTT Center

- *Goal:* To enable high-fidelity calculations based on multiple coupled physical processes and multiple physical scales
  - *Adaptive methods*
  - *Composite or hybrid solution strategies*
  - *High-order discretization strategies*

- *Barrier:* The lack of easy-to-use interoperable meshing, discretization, and adaptive tools requires too much software expertise by application scientists

*The TSTT center recognizes this gap and will address the technical and human barriers preventing use of adaptive, composite, hybrid methods*

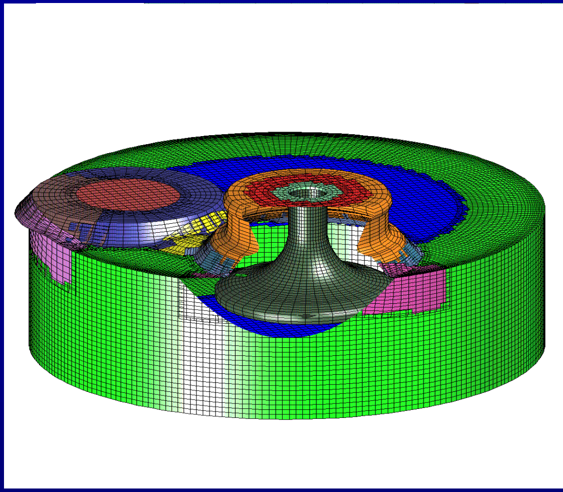
# TSTT Participants

- ANL: *Fischer*, Leurent
- BNL: *Glimm*, Oh, Samulyak
- LLNL: *Brown*, Chand, Henshaw, Quinlan, White
- ORNL: *D' Azevedo*, de Almeida, Khamayseh
- PNNL: *Trease,* Trease
- RPI: Datta, Flaherty, Kokak, Luo, Seole, *Shephard*
- SNL: Brewer, Freitag, *Knupp*, Melander, Tautges
- SUNY SB: Glimm, *Li,* Miller

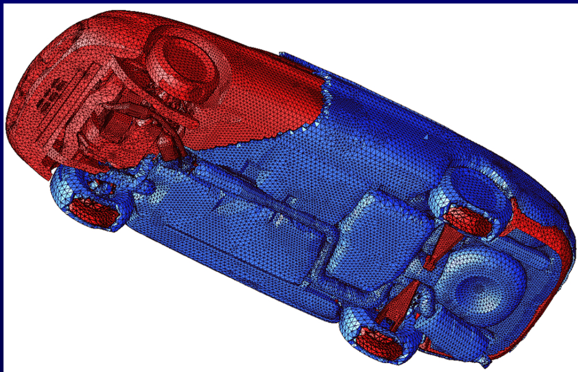
**Italics denote site PI*

# TSTT brings together meshing and discretization expertise from DOE Labs and Universities

- **Structured meshes**
  - **Overture** - high quality, predominantly structured meshes on complex CAD geometries (LLNL)
  - Variational and Elliptic Grid Generators (ORNL, SNL)
- **Unstructured meshes**
  - **AOMD** (RPI) - primarily tetrahedral meshes, boundary layer mesh generation, curved elements, AMR
  - **CUBIT** (SNL) - primarily hexahedral meshes, automatic decomposition tools, common geometry module
  - **NWGrid** (PNNL) - hybrid meshes using combined Delaunay, AMR and block structured algorithms
  - **Frontier** (BNL) – interface front tracking



*Overture* Mesh (LLNL)



*MEGA* Boundary Layer Mesh (RPI)

# Bringing this sophisticated technology to DOE application scientists is the challenge

- These tools all meet particular needs, but
  - They do not interoperate to form hybrid, composite meshes
  - They cannot be easily interchanged in an application

- In general the technology requires too much software expertise from application scientists
  - Difficult to improve existing codes
  - Difficult to design and implement new codes

# We meet this challenge through a 2-pronged approach

Near term collaborations helps us understand application requirements …

- *Near term:* deployment of current TSTT mesh and discretization capabilities by partnering with SciDAC applications
- *Long term:* development of interoperable software tools enabling
  - Rapid prototyping of new applications
  - Plug-and-play insertion of mesh and discretization technology through uniform software interfaces

… feeding into interface design of future software components

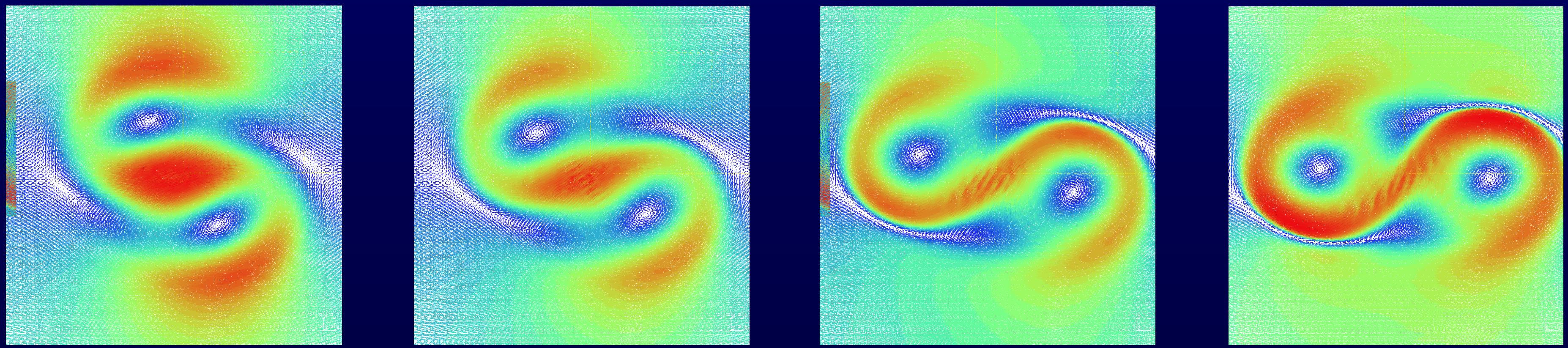
# Near Term Strategy

- Interact with SciDAC Applications to develop working relationships in each application area by
  - Analyzing the needs of application scientists
  - Using existing tools and technologies to prototype and demonstrate new solution strategies
  - Inserting existing TSTT technology
    - Provides a short-term impact for application scientists
    - Builds trust relationship
  - Developing new technologies for later insertion and new application development

- Key application areas: Fusion, Astrophysics, Accelerator Design, Climate

# High-Order FEM for Fusion

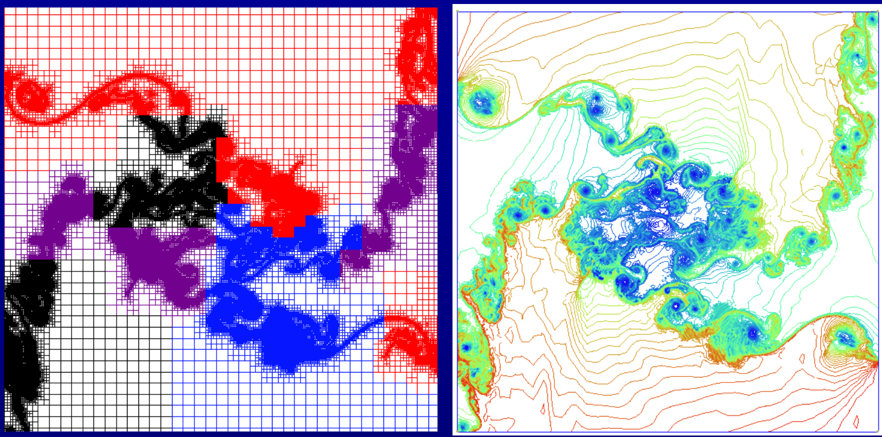
- High-order, adaptive finite element techniques for magneto-hydrodynamics
  - Fusion PI: Jardin/Strauss (PPPL)
  - TSTT PI: Shephard/Flaherty (RPI)
  - *Goal:* To test high-order and adaptive techniques; compare to existing linear FEM
  - *Progress:*
    - Initial results obtained for both potential and primitive variable mixed formulations for the 2D adipole vortex flow pattern
      - Two oppositely directed currents embedded in a constant magnetic field which holds them in an unstable equilibrium
      - They compress and rotate to align with magnetic field to reduce energy (see below)
    - Testing high-order and h-adaptive techniques available in Trellis to determine applicability to this problem
    - Quadratic and cubic results presented by J.E. Flaherty at SIAM Annual Meeting 02



SciDAC
Scientific Discovery through Advanced Computing

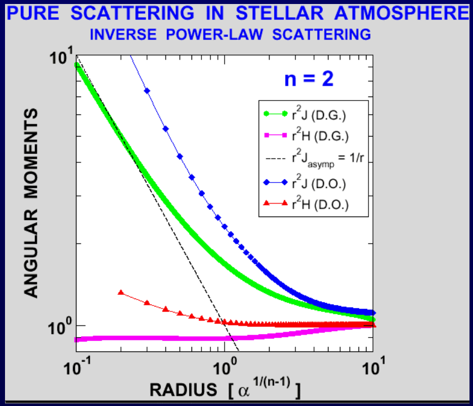
# Adaptive DG for Astrophysics

- Contact instabilities in hydrodynamics
  - Application PI: Bhattacharjee/Rosner (Iowa/UofC)
  - TSTT PI: Shephard (RPI)
  - *Goal:* to test h-p adaptive DG in hydrodynamics; compare to existing PPM
  - *Progress:* 3-D adaptive test to 256 processors have been done in Trellis for four contact Riemann problem



Adaptive mesh and density contours after structures have evolved. Colors on right mesh indicates processor assignment for this 4 processor case

- Boltzman transport equations for neutrinos
  - Application PI: Mezzacappa (ORNL)
  - TSTT PI: de Almeida (ORNL)
  - *Goal:* to compare adaptive DG and discrete ordinates discretization (non-adaptive, computationally intensive)
  - *Progress:* adaptive DG shows strong exponential decay, energy conservation, and outward peaking and gives comparable results faster than Discrete Ordinates in 1D test case
  - Need to extend to more complex test cases



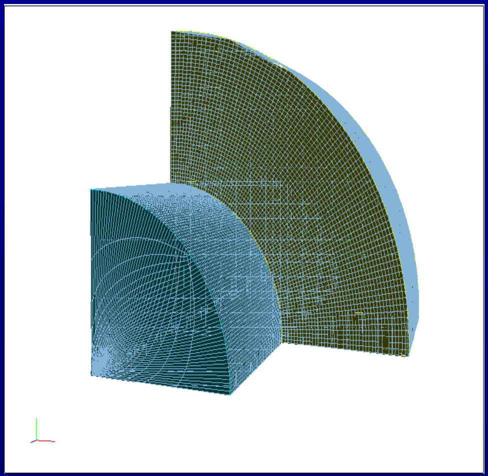

Mean Radiation Intensity (J); Net Energy Flux (H)

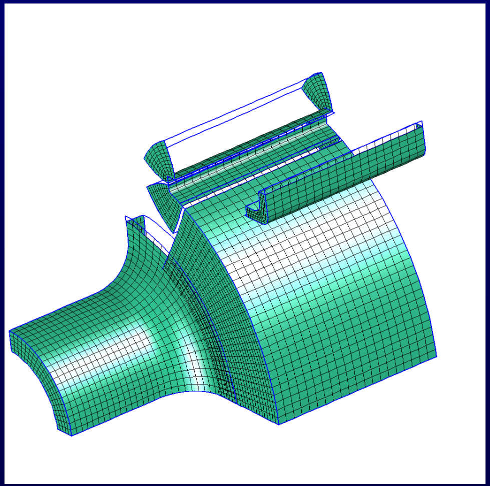
DOM does not reach asymptotic limit at large optical depth and does not conserve energy

Adaptivity in DGM provides more accuracy the slight loss of energy will be corrected

# Mesh Quality in Accelerator Design

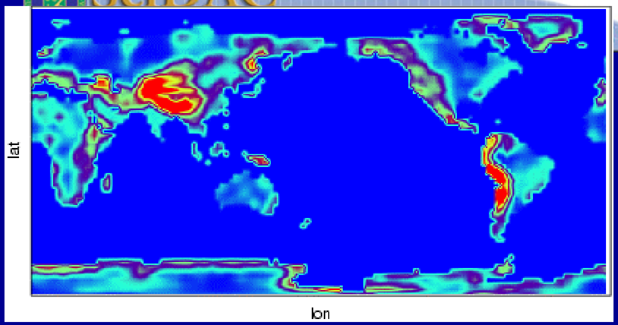
- Understanding the effect of mesh quality on Tau3P
  - Application PI: Ko/Folwell (SLAC)
  - TSTT PI: Knupp (SNL), Henshaw (LLNL)
  - *Goal:* Determine the mesh quality factors that most affect stability of Tau3P and to devise discretization schemes to improve the stability of Tau3P without affecting long-time accuracy
  - *Progress:*
    - Systematic mesh quality analysis using CUBIT meshes revealed that run time varies by a factor of 3 from "best" to "worst" mesh and that smoothness and orthogonality are the most important factors
    - Analytically derived sufficient conditions on mesh quality for stability of discretization in Tau3P
    - Implemented basic Tau3P discretization strategy in Overture and analyzing feasibility of schemes stabilizing the DSI method
    - Demonstrating high order FEM methods for next generation code T3P

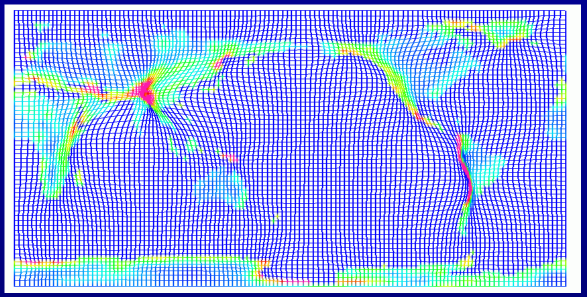


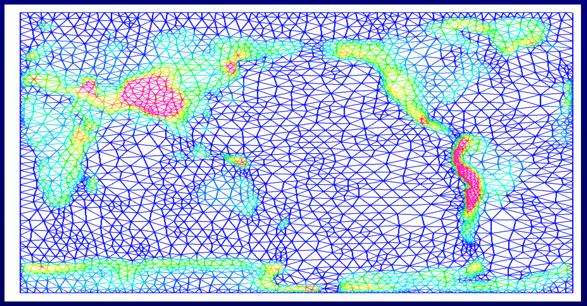


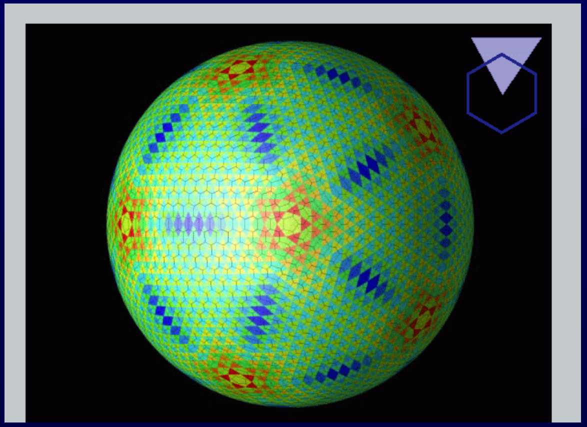
# Climate

- Adaptive gridding to minimize solution error
  - Application PI: Drake (ORNL)
  - TSTT PI: Khamayseh (ORNL)
  - *Goal:* Given an initial isotropic or anisotropic planar or surface mesh and a solution field with large gradient mountain heights, use solution based r-adaptation to minimize solution error
  - *Progress*: Proof of principle of meshing technologies demonstrated, used in next generation climate codes
- Geodesic mesh quality improvement
  - Application PI: Randall/Ringler (Colorado)
  - TSTT PI: Knupp (SNL)
  - *Goal:* Create smoothed geodesic grids to improve calculation accuracy
  - *Progress:* Used early version of Mesquite to create smoothed grids with respect to element area and perform calculations with smoothed grids to determine effect on accuracy



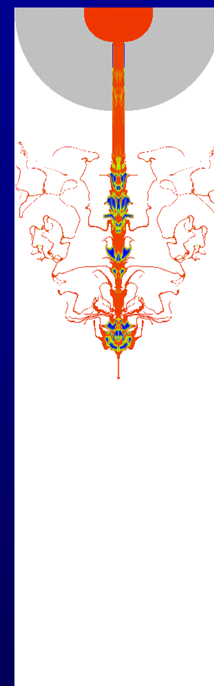



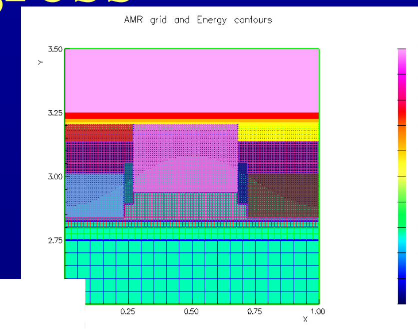


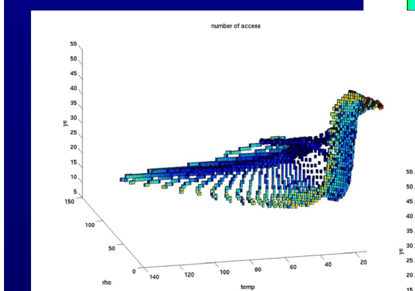


# Other examples where TSTT technology is helping near-term application progress

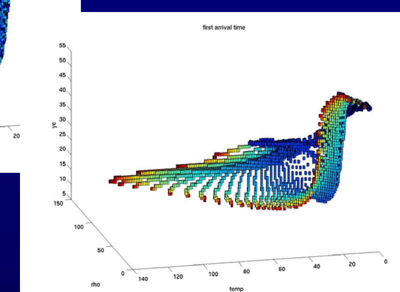
- Front tracking and adaptive techniques in Frontier and Overture for modeling of the breakup of a diesel fuel jet into spray (Argonne/BNL)
- 3D caching schemes to avoid redundant, costly evaluations of scattering kernels in phase space in astrophysics calculations (ORNL/ORNL)
- Mesh-based schemes for computational biology applications such as rat olfactory systems and human lungs (PNNL/PNNL)
- Low-order FEM schemes used as effective preconditioners in Climate applications (Colorado/ANL)



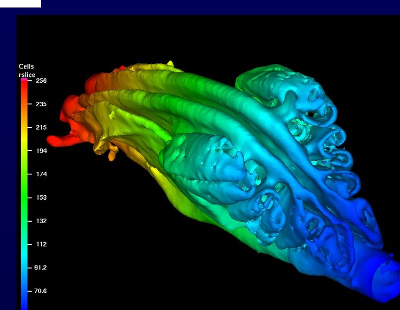


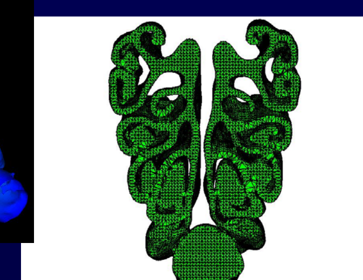




Access pattern :red is more frequent




Access sequence: red is accessed late in simulation



# Long Term Strategy

- Create interoperable meshing and discretization components
  - Common interfaces for mesh query and modification
  - Initial design will account for interoperability at all levels
  - Encapsulate existing TSTT software tools into CCA-compliant components for plug and play
- Develop new technologies as needed to enable interoperability
  - High-level discretization library
  - Mesh quality improvement technologies
  - Terascale algorithms for adaptivity, load balancing, interpolation

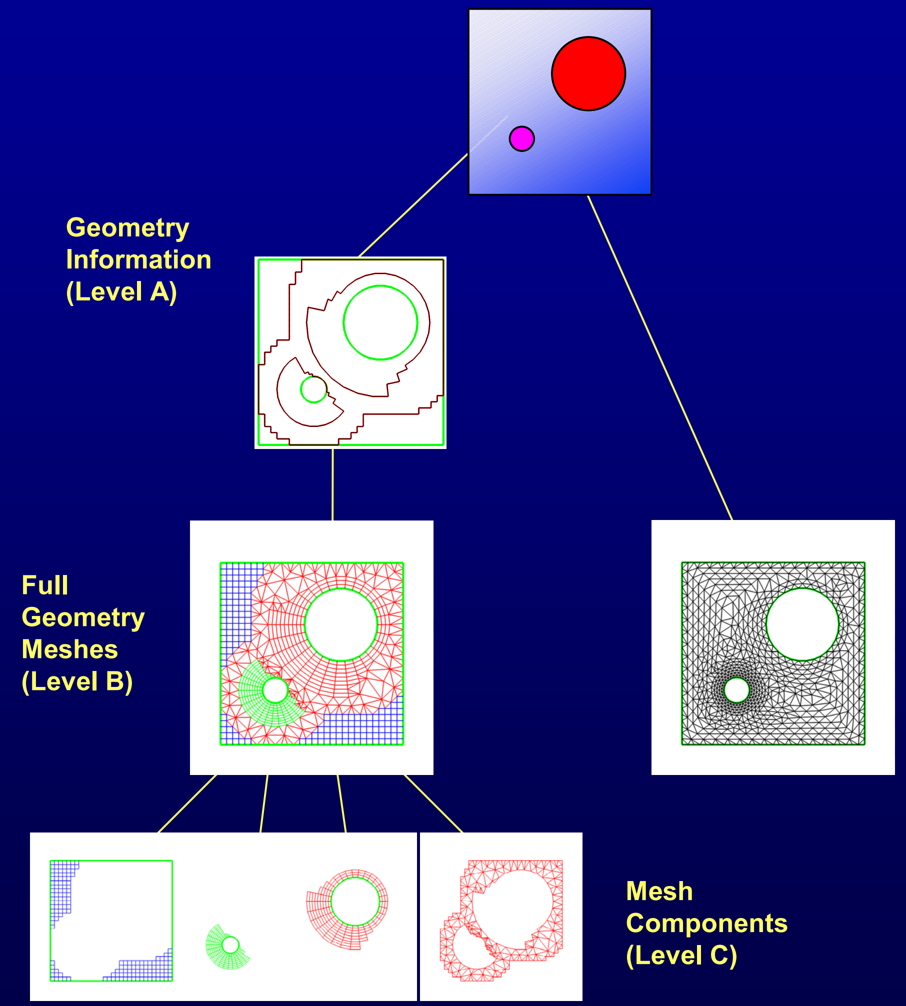
# TSTT Interoperability Goal

To provide *interchangeable* and *interoperable* access to different mesh management and discretization strategies

- Ease experimentation with different technologies
- Combine technologies together for hybrid solution techniques

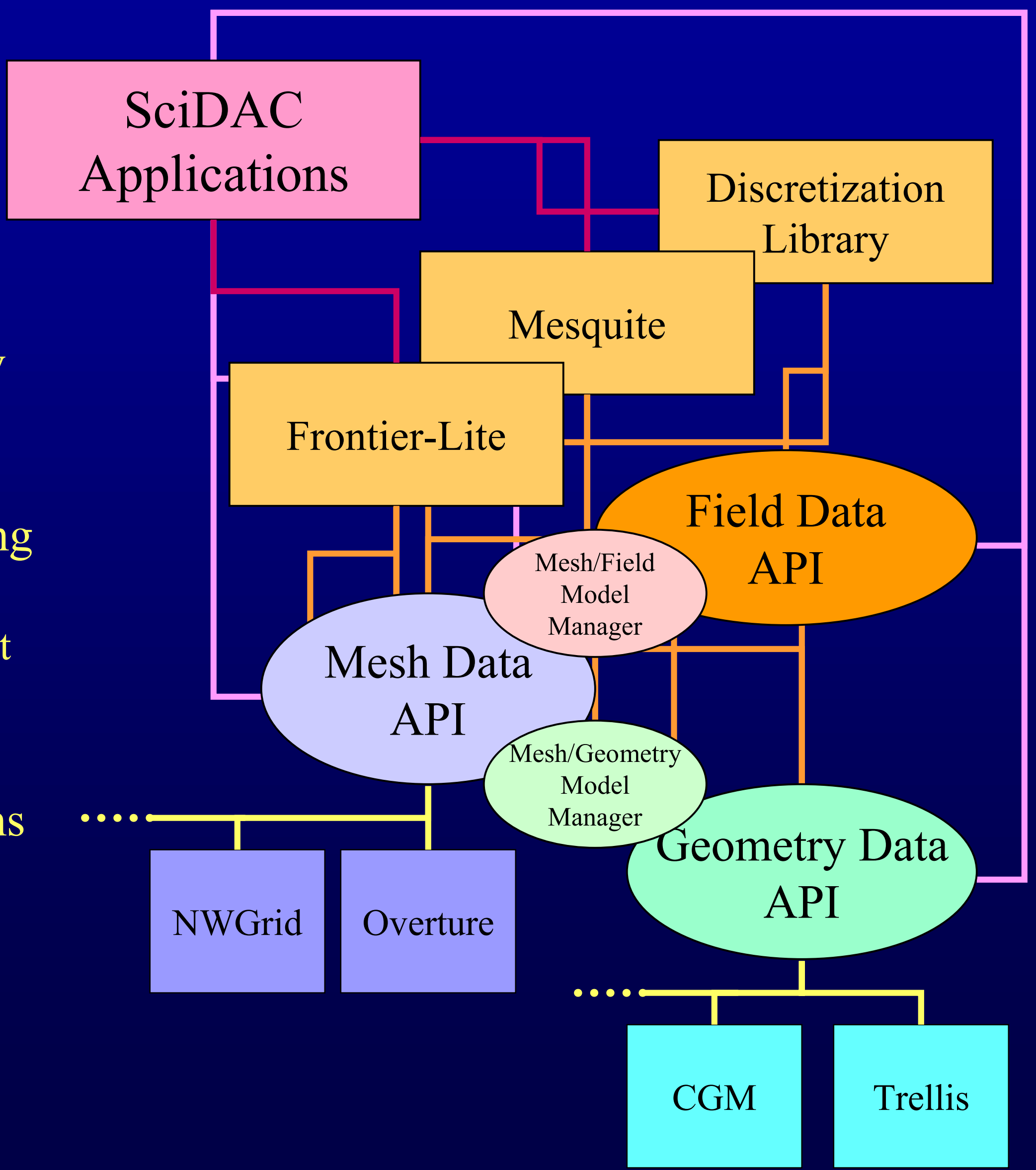
The Data Hierarchy

- **Level A: Geometric description of the domain**
  - provides a common frame of reference for all tools
  - facilitates multilevel solvers
  - facilitates transfer of information in discretizations
- **Level B: Full geometry hybrid meshes**
  - mesh components
  - communication mechanisms that link them (key new research area)
  - allows structured and unstructured meshes to be combined in a single computation
- **Level C: Mesh Components**



# Interoperability Development Plan

- Define interfaces for
  - Mesh Data
  - Geometry Data
  - Field Data
  - Data Model Managers
- Wrap existing TSTT tools to comply with these interfaces
- Create new tools that use these interfaces to work with the underlying infrastructures interchangeably
  - Mesquite mesh quality improvement
  - Discretization Library
  - Frontier-Lite
- Use these tools to impact applications
- Use TSTT interfaces directly in applications
- Use TSTT tools interoperably



# Interface Definition Philosophy

- Create a small set of interfaces that existing packages can support
  - Very small set of 'core' functions that must be implemented
  - Larger set of functions supported by reference implementations
- Balance performance and flexibility
- Work with a large tool provider and application community to ensure applicability
- Enable both interchangeability and interoperability
- Use CCA technologies as appropriate
  - SIDL/Babel for language interoperability
  - Some use of Ccaffiene and Decaf frameworks for developing prototype TSTT components

# TSTT Data Model

- Core Data Types
  - *Mesh Data*: provides the geometric and topological information associated with the discrete representation of the computational domain
  - *Geometric Data*: provides a high level description of the boundaries of the computational domain; e.g., CAD, image or mesh data
  - *Field Data*: *(preliminary)* the time dependent physics variables associated with application soluiton.  These can be scalars, vectors, tensors, and associated with any mesh entity.
- Data Relation Manager
  - Provides control of the relationships among two or more the core data types.  It resolves cross references between entities in different groups and provides additional functionality that depends on multiple core data types.

# Data and Format

- Infrastructure:
  - Handles: An opaque object that represents entities, entity sets, tags to the application
  - Tags: User-defined opaque data with a unique string name, size (in bytes), tag handle
  - Errors
- Mesh Data:
  - Entities
  - Entity Sets
  - Meshes (Static, Modifiable)
- Geometry Data
- Mesh/Geometry Interface
- In each case there is
  - A 'definition',
  - Expected capabilities
  - Examples
  - Functional interfaces

*Use this as a detailed example*

# Mesh Entities

- Definition
  - Unique type and topology
    - Type: Vertex, Edge, Face, Region
    - Topology: Point, Line_Segment, Polygon, Triangle, Quadrilateral, Polyhedron, Tetrahedron, Hexahedron, Prism, Pyramid, Septahedron
  - Faces and regions have no interior holes
  - Higher-dimensional entities are defined by lower-dimensional entities through canonical ordering relationships
- Capabilities
  - Return upward and downward first order adjacencies in the canonical ordering
  - Support both individual and agglomerated request mechanisms
  - Vertices return coordinate information in arrays of doubles
  - Add, retrieve, set, and delete user defined tag data
- Examples
  - Vertex (0D), edge (1D) , triangular face (2D), tetrahedral region (3D)

# Mesh

- Definition
  - A collection of TSTT entities that have uniquely defined entity handles
  - Entities are related through topological adjacency information in which higher-dimensional entities are defined by lower dimensional entities
    - This definition may or may not be unique
- Examples
  - Type 1: a non-overlapping connected set of TSTT entities, e.g., a conformal finite element mesh
  - Type 2: a collection of Type 1 meshes used to represent the computational domain. These may or may not be overlapping meshes
  - Adaptive meshes in which both coarse and fine TSTT entity regions are retained in the database. The most highly refined regions of this mesh typically comprises a Type 1 or Type 2 mesh
  - SPH meshes which consist of a collection of vertices

# Mesh Capabilities

- Static Type 1 Meshes
  - Populating the interface by string name
  - Basic query capabilities
    - Entities and adjacency information
    - Array or iterator-based
  - Add, retrieve, set and delete user-defined tags
- Extensions
  - Subsetting to create arbitrary groupings of mesh entities
  - Modifiable meshes
  - Connections to geometric entities

# Entity Sets

- Definition
  - Arbitrary groupings of TSTT mesh entities
    - May or may not be a multiset or ordered
    - May or may not be a valid computational mesh
  - Multiple entity sets can be associated with a given mesh
  - Relationships between entity sets
    - Contained in (subset) relationship
    - Parent/Child relationship
    - Default on creation is *contained in* parent mesh
- Capabilities
  - Static Type 1 capabilities as before
  - Set Operations
    - Add and remove existing TSTT entities to the mesh set
    - Add, subtract, intersect, or union entity sets
    - Subset and Parent/Child relationships
- Examples
  - A set of vertices, the set of all faces on a geometric face, the set of regions in a domain decomposition for parallel computing

# Issues that have arisen

- Nomenclature is harder than we first thought
- Cannot achieve the 100 percent solution, so...
  - What level of functionality should be supported?
    - Minimal interfaces only?
    - Interfaces for convenience and performance?
  - What about support of existing packages?
    - Are there atomic operations that all support?
    - What additional functionalities from existing packages should be required?
  - What about additional functionalities such as locking?
- Language interoperability is a problem
  - Most TSTT tools are in C++, most target applications are in Fortran
  - How can we avoid the “least common denominator” solution?
  - Exploring the SIDL/Babel language interoperability tool
- Performance is a critical aspect
- The devil is in the details

# TSTT Interface Status

- Implementations
  - RPI, LLNL, SNL/ElemTech, ANL, PNNL
- Immediate uses of TSTT interfaces
  - Mesquite mesh quality improvement toolkit (ANL, SNL)
  - VERDE quality assessment (SNL, ElemTech)
  - Frontier interoperability with mesh generation codes (BNL, LLNL, PNL)
  - VTK visualization (LLNL)

– Working closely with the CCA Babel team to evaluate performance of TSTT interfaces

# High Level Access

- Operate on the mesh components as though they were a single mesh object
  - Discretization operators
  - Mesh modifications
    - Mesh quality improvement
    - Refinement/coarsening
  - Error estimation
  - Multilevel data transfer
- Prototypes provided by Overture and Trellis frameworks
- Enables rapid development of new mesh-based applications

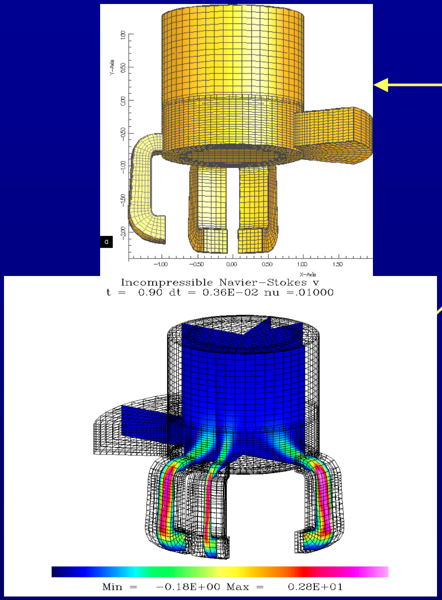
# Discretization Library

- *Observation:* Complexities of using high-order methods on adaptively evolving grids has hampered their widespread use
  - Tedious low level dependence on grid infrastructure
  - A source of subtle bugs during development
  - Bottleneck to interoperability of applications with different discretization strategies
  - Difficult to implement in general way while maintaining optimal performance
- Result has been a use of sub-optimal strategies or lengthy implementation periods
- *TSTT Goal:* to eliminate these barriers by developing a *Discretization Library*

# Functionalities

- Mathematical operators will be implemented
  - Start with +, -, *, /, interpolation, prologation
  - Move to div, grad, curl, etc.
  - Both strong and weak (variational) forms of operators when applicable
- Many discretization strategies will be available
  - Finite Difference, Finite Volume, Finite Element, Discontinuous Galerkin, Spectral Element, Partition of Unity
  - Emphasize high-order and variable-order methods
  - Extensive library of boundary condition operators
- The interface will be independent of the underlying mesh
  - Utilizes the common low-level mesh interfaces
  - All TSTT mesh tools will be available
- Interface will be extensible, allowing user-defined operators and boundary conditions

# Example provided by Overture prototype



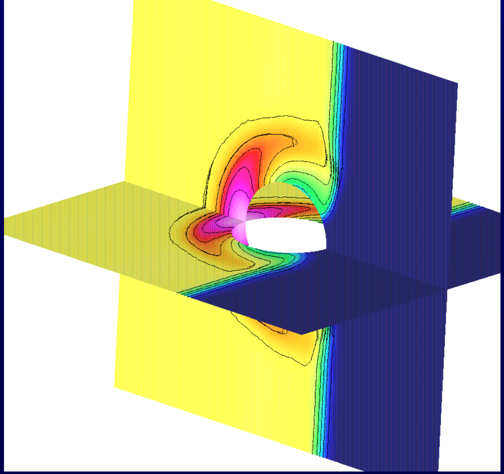

```
CompositeGrid cg;
floatCompositeGridFunction u,v,w;
```

```
v = u.y();
w = u.laplacian();
```

*Differentiation Operators*

```
Plotstuff ps;
ps.plot (cg);
ps.contour (w);
```

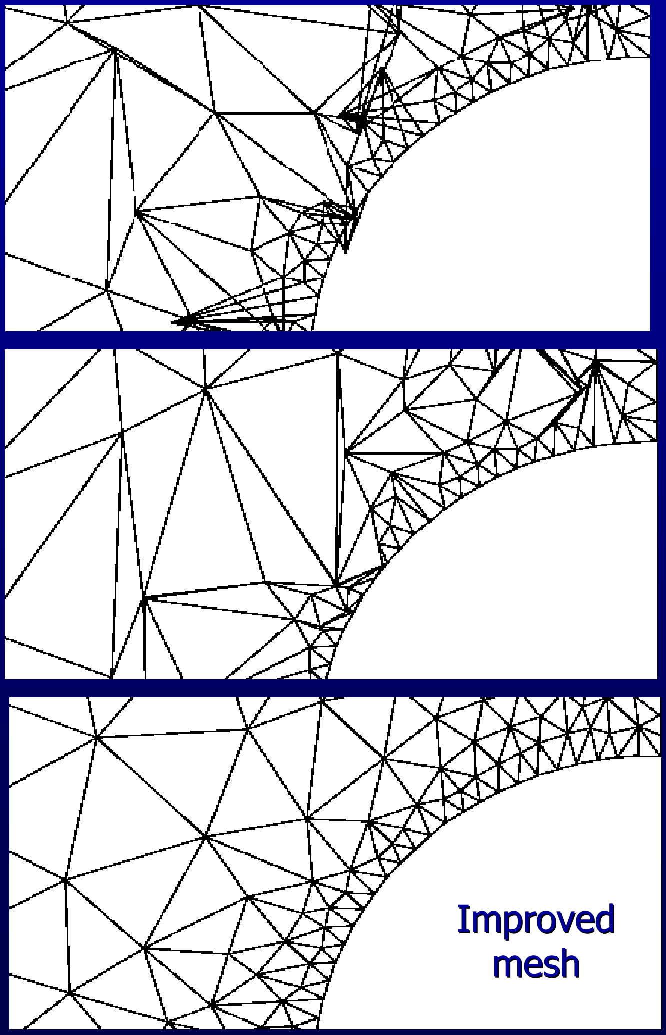
*Visualize grid and data*



**Trellis (RPI) provides similar capability for finite-element method**

# MESQUITE Mesh Quality Improvement

- *Goal*: To provide a stand-alone tool for mesh quality improvement
  - hybrid, component based meshes
  - development of quality metrics for high order methods
  - a posteriori quality control using error estimators
- *Team*
  - Micheal Brewer (SNL)
  - Lori Freitag Diachin (SNL)
  - Patrick Knupp (SNL)
  - Thomas Leurent (ANL)
  - Darryl Melander (SNL)



# Mesh Improvement Strategies

- Goals
  - a priori shape, size, alignment improvement
  - a posteriori solution improvement
- Methods
  - Vertex repositioning
    - Laplace smoothing
    - PDE-based solvers
    - Numerical optimization schemes
  - Topology modifications
    - Face and edge swapping
  - *h*-refinement

*There exists no stand alone software toolkit that addresses mesh quality improvement for a broad range of mesh element types and improvement strategies*

# MESQUITE Vision

- Provide a comprehensive, stand-alone toolkit for mesh quality improvement with the following capabilities
  - Shape Quality Improvement
  - Mesh Untangling
  - Alignment with Scalar or Vector Fields
  - R-type adaptivity to solution features or error estimates
    - Maintain Quality of Deforming Meshes
    - Anisotropic Smoothing
    - Control Skew on Mesh Boundaries

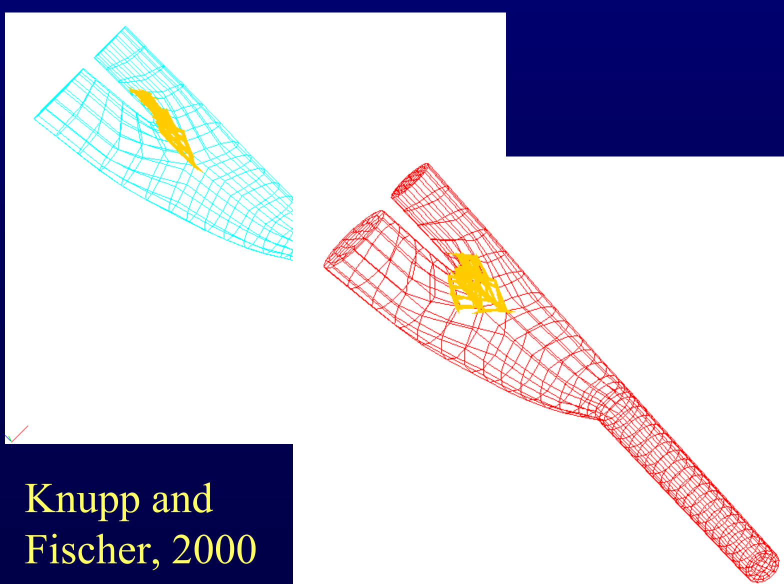
# Example of Mesh Improvement Impact

- Arteriovenous Graft Turbulent Flow Simulation

- Compute maximum shear stress with high order spectral methods
  - Poorly-shaped Elements Increase CG Solver Iterations

- Mesh Optimized by Condition Number
  - reduced maximum number of solver iterations from 169 to 150
  - reduced the average from 18.06 to 15.46 (about a 17% savings).

*Four hours of Applications Solver time was traded for 19 minutes of mesh smoothing time.*

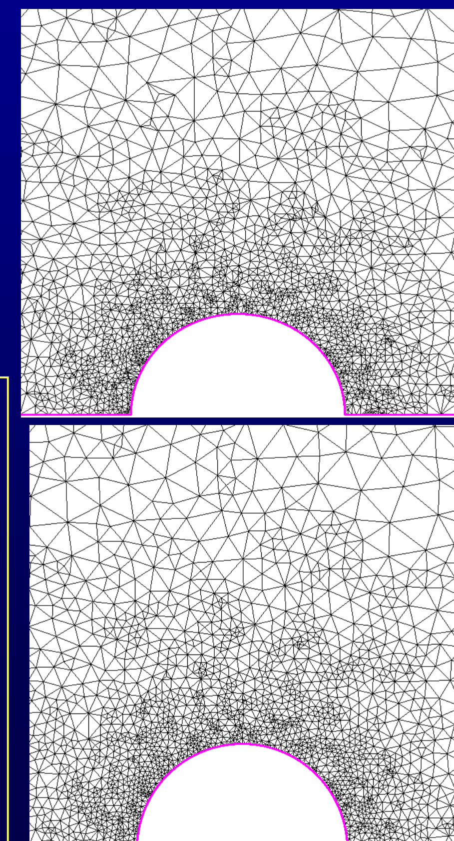


Knupp and Fischer, 2000

- Compressible Flow

- Mesh Optimized w/ Active set solver
  - Improved the convergence rate by 25%

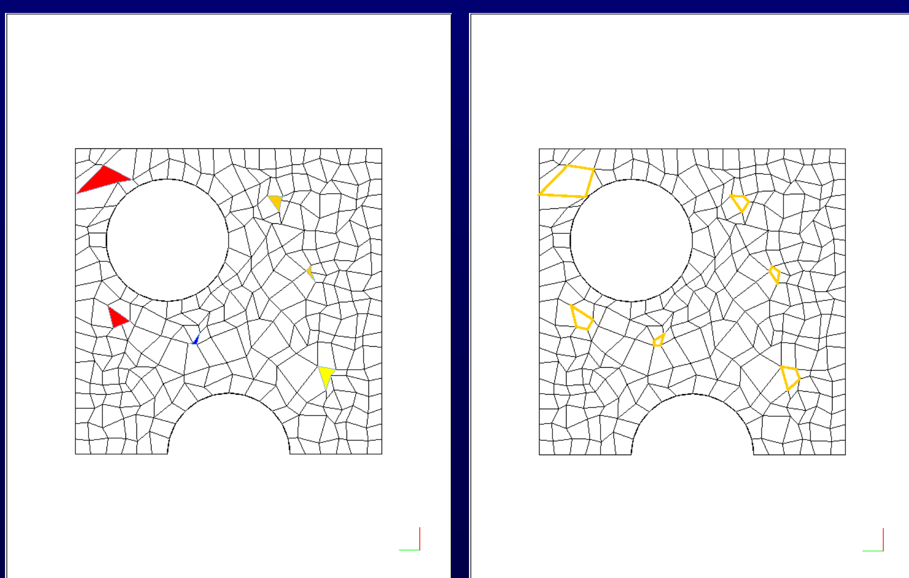
- Mesh improvement cost less than one multigrid iteration



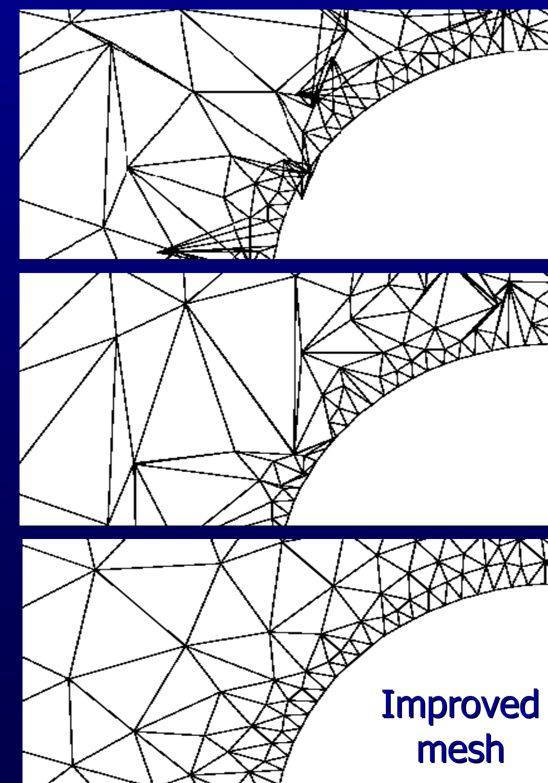
Freitag and Ollivier-Gooch, 1998

# Impact of Mesh Untangling

- Few hex-meshing algorithms guarantee the quality of the mesh
  - Inverted elements are produced
- Mesh untangling algorithms can remove inverted elements quickly
  - Eliminates need to remesh
  - Eliminates the need to re-decompose the geometry



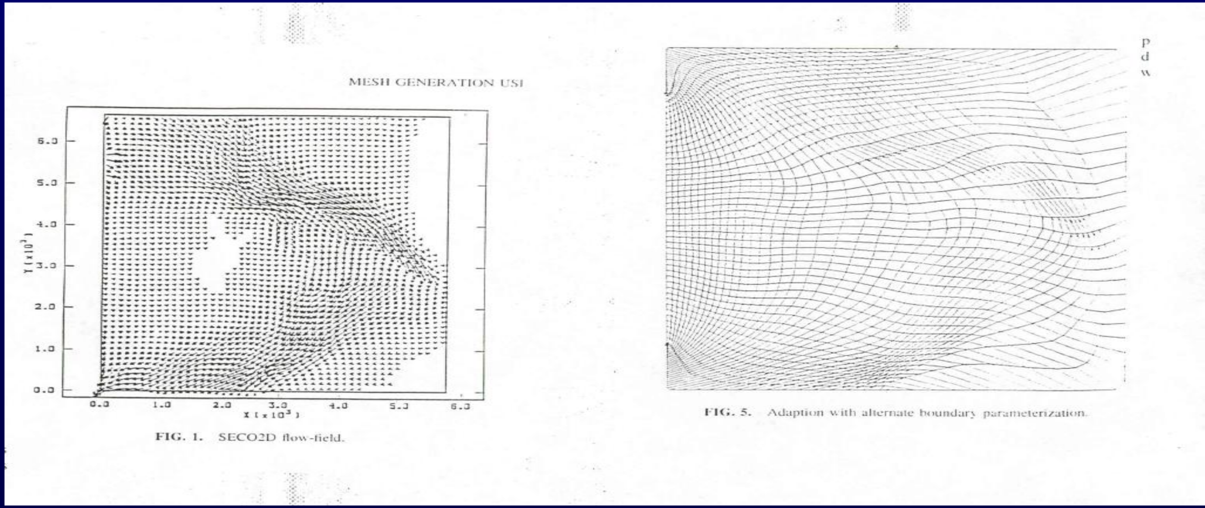
Knupp



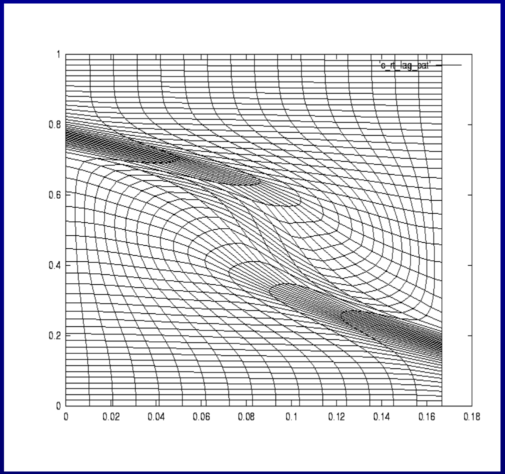

Freitag

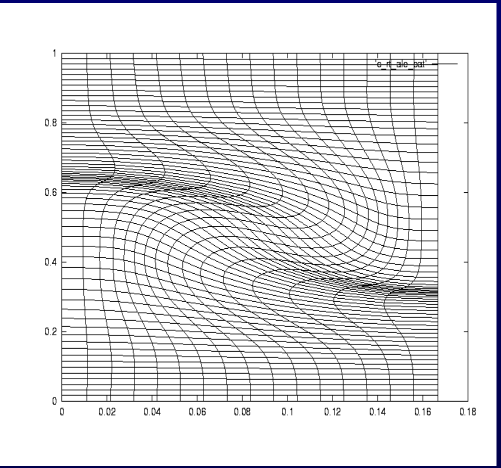
# Mesh Alignment

- Moving vertex positions to match a vector or scalar field
- Improving ALE mesh quality while preserving flow characteristics
- Deforming a mesh to match a perturbed geometrical domain




Knupp, 1996





Knupp, Shashkov, Garimella 2000

# Mesquite Capabilities

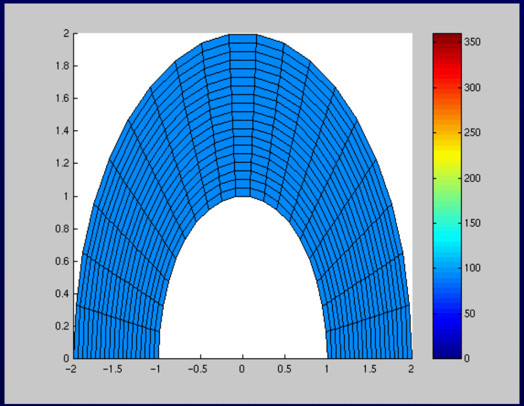
- Problem Domain of Interest
  - Structured, *Unstructured, Hybrid*, and NonConforming Meshes
  - 1D*, 2D, 3D*, curve, *surface*, *volume*
  - *Hex, tet*, pyramid, prismatic, polyhedral, high-order elements
  - Adaptive & non-adaptive applications
- Technologies
  - *Node movement algorithms*
  - Local topology modifications
  - Constrained/*unconstrained optimizations*
  - *Numerical optimization* & PDE-based solvers
- Previous Experience
  - CUBIT mesh improvement algorithms (P. Knupp PI)
  - Opt-MS mesh improvement algorithms (L. Freitag PI)

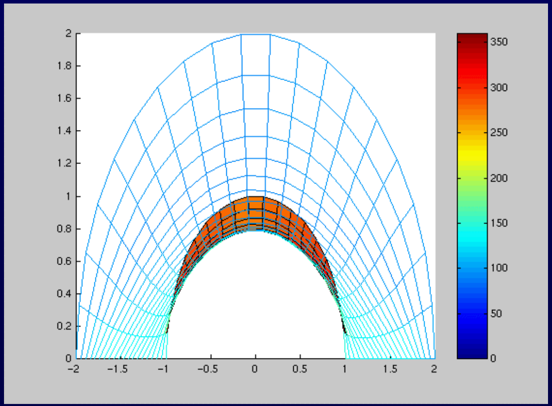
_*Italics denotes existing capabilities_

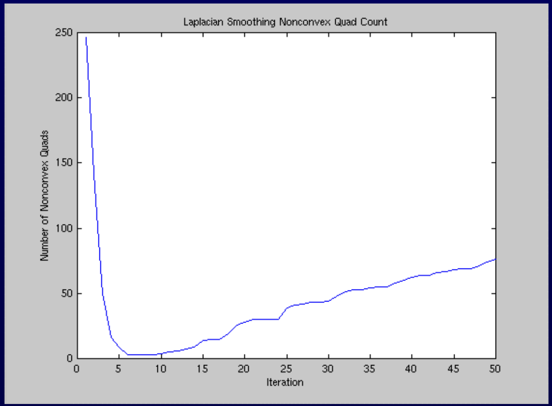
# Smoothing Algorithms

## Laplacian Smoothing

- Move the free vertex to the geometric center of the adjacent vertices
- Quality improvement is *not* guaranteed
  - Can result in invalid, or tangled, meshes
- Computationally inexpensive
- Easy to implement
- Best used as a preprocessing step to optimization-based techniques







# Optimization-based Smoothing Techniques

- Comprised of quality metrics, objective functions and solution algorithms
- Quality metrics, $q_i$
  - *A priori* geometric criteria
    - Ratio of volume to face areas (e.g., Shephard and Georges, 1991, Bank 1994)
    - Angle-based and other geometric measures (e.g., Freitag, et al. 1995)
    - Distortion metrics (e.g, Canaan, 1998)
    - Element condition number and other matrix norms (Knupp, 1999)
  - *A posteriori* local error analysis (e.g., Bank and Smith, 1997, Berzins, et. al., 2000 )
- Objective functions
  - Minimize the average $q_i$ ($L_2$ norms)
  - Minimize the maximum $q_i$ ($L_\infty$ norms)

# Optimization-based Smoothing Techniques

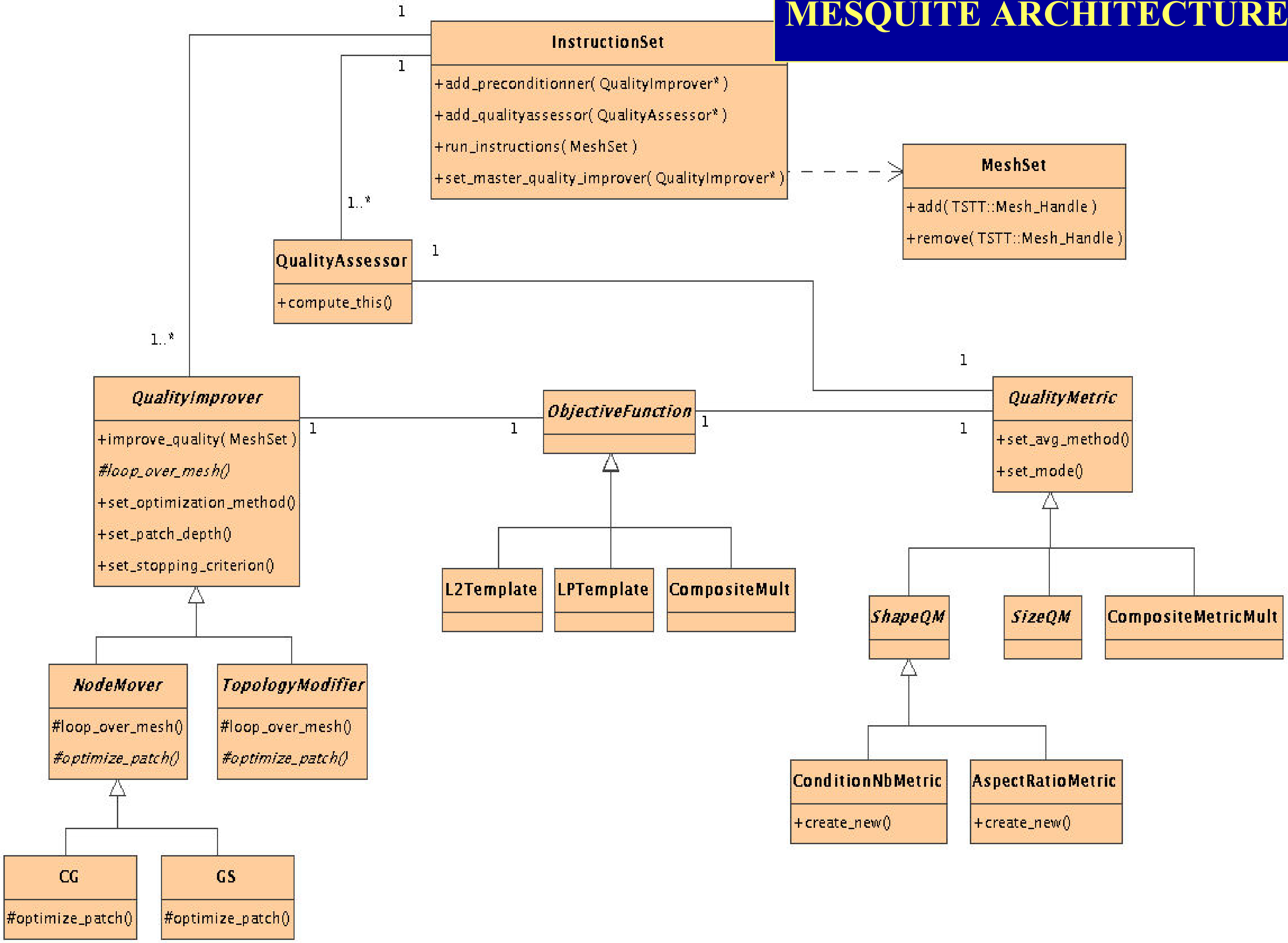
- Optimization methods
  - Steepest descent active set methods (Freitag, 1995, Amenta, et. al. 1996, Canaan, 1998)
  - Nonlinear conjugate gradients (Knupp, 1998)
  - Feasible Newton methods (Munson, et. al. 2001)
  - Combination approaches (Shephard and Georges 1991, Freitag 1997, Freitag and Knupp 1999)

- Design space
  - Local: relocate a single vertex and sweep through the mesh
  - Global: relocate all vertices simultaneously

# Mesquite Software Design Principles

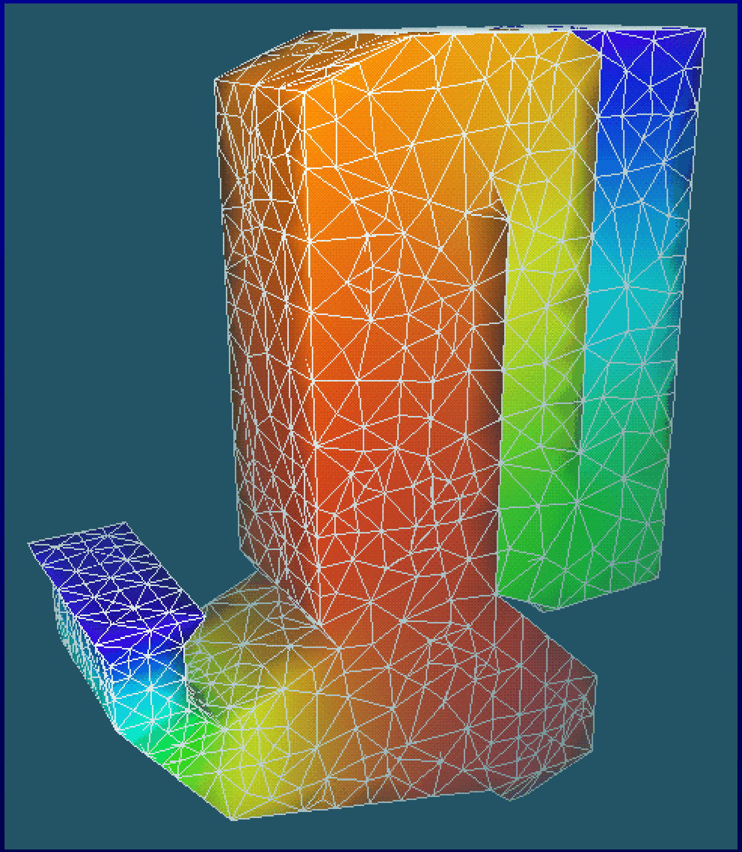
- Object oriented software
  - Objects correspond to mathematical abstractions
  - Use well-defined interfaces for interactions with mesh and geometry
- Provide automatic mesh improvement strategies and simple interfaces for ease of use
- Allow customization
  - Mix and match flexibility
  - User-set stopping criterion parameters

# MESQUITE ARCHITECTURE



# Current Software Status

- Quality Metrics
  - Condition Number
  - Mean Ratio
  - Aspect Ratio
  - Untangling
- Objective Function Templates
  - $L_2$ and $L_P$
  - Minimum $L_\infty$
- Vertex Movers
  - Steepest Descent
  - Nonlinear Conjugate Gradients
  - Active Set Solvers
  - Laplace smoothers
- Solution Domain
  - Local
  - Global



# Accessing Information

- **Mesh Information**
  - The TSTT mesh query interface
    - Mesquite currently working with AOMD and MDB implementation
    - Upgrading to the latest TSTT spec using SIDL/Babel
  - Mesquite mesh query interface
    - Stand-alone C++ abstract classes
    - Less broad than the TSTT spec definition
    - Data neutrality using handles a la the TSTT spec
- **Geometric model**
  - MESQUITE will not have its own geometry engine
  - Some simple call-back functions such as “move to owner” and “surface normal” by the application or TSTT
  - TSTT developing a common interface for this functionality

# MESQUITE User Interface

- **Multi-level API**
  - Simple to use wrapper interface
    - Access Mesquite functionality in a minimal number of calls
    - Uses default algorithms, settings, stopping criterial
  - Low level interface for customization
    - User chooses the combination of metric, objective function, solver
    - User determines the instruction queue
- **Assessment Tools**
  - Diagnostics
  - Statistics
  - A priori and a posteriori quality assessment
- **Users' Manual and Documentation**

# Mesquite User Interface: Wrappers

```
#include Mesquite.h

void some_application_function{

   TSTT_Mesh tri_mesh, quad_mesh;

   Mesquite::initialize();

  // create a Mesh Set to hold the TSTT meshes
  Mesquite::MeshSet ms;
  ms->add_mesh(tri_mesh);
  ms->add_mesh(quad_mesh);

  // state the improvement objective
  ShapeQualityObjective shape_quality_objective;

  // improve the quality
  shape_quality_objective.improve_quality(ms);

  Mesquite::finalize();
}
```

# Mesquite User Interface: Customized

```
void some_application_function{

   TSTT_Mesh tri_mesh, quad_mesh;

   Mesquite::initialize();

  // create a Mesh Set to hold the TSTT meshes
  MeshSet ms;
  ms->add_mesh(tri_mesh);
  ms->add_mesh(quad_mesh);

  ShapeQualityMetric *condition_number_metric = Conc

  ObjectiveFunction *shape_objective_function = new L2

  QualityAssessor *shape_quality_assessor = new QualityAssess
  shape_quality_assessor->compute_this(QualityAssessor::Mini

  NodeMover *opt_L2 = new NodeMover(shape_objective_function);
  opt_L2 = set_optimization_method(NodeMover::FeasibleNew
  opt_L2 = set_stopping_criterion(MAX_NODE_MOVEMENT
```

Initialize

Declare a shape quality metric

Declare an objective function

Create a quality assessor

Create a L2 Node Mover

# Mesquite User Interface

```
TopologyModifier *tet_swapper = new TopologyModifier(o
tet_swapper->set_optimization_method(TopologyModifier:
tet_swapper->set_stopping_criterion(MESH_PASSES,1);

UntangleQualityMetric *untangle_metric = FirstUntangleMetric::create_new();
ObjectiveFunction *untangle_objective_function = new LINF_TEMPLATE(untangle_metric);
NodeMover *opt_LINF = new NodeMover(untangle_
opt_LINF->set_optimization_method(NoveMover::Si
opt_LINF->set_stopping_criterion(OBJ_FCN_VAL,0

InstructionQueue q;
q.add_quality_assessor(shape_quality_assessor);
q.add_preconditioner(opt_LINF);
q.add_preconditioner(tet_swapper);
q.add_master_quality_improver(opt_L2);
q.add_quality_assessor(shape_quality_assessor);
q->execute_instruction_queue(ms);

Mesquite::shutdown();

}
```

Create a Topology Modifier

Create an untangler (metric, Objective function, NodeMover)

Create an instruction queue

Execute the instruction queue

Finalize and shutdown

# User Customization

- Users can insert their own algorithms, objective functions, quality metrics without recompiling Mesquite
  - Inherit from VertexMover, ObjectiveFunction or QualityMetric
- User-defined metrics/objective functions can take advantage of existing MESQUITE algorithms
- Provides a platform for new research in mesh improvement algorithms
- Provides a platform for comparative studies

# Achieving Efficiency

- **Algorithmic**
  - State-of-the-art optimization algorithms
  - Mesh preconditioners such as constrained Laplacian smoothing
  - Flexible stopping criterion
  - Pruning techniques
- **Coding Practices**
  - Outer layers coded in C++ for Maintainable Code
  - Inner kernel will be C, arrays, in-lined functions for speed
- **Parallel Computing (proposed)**
  - Partitioning strategies for large meshes
  - Parallel algorithms for global techniques

SciDAC Scientific Discovery through Advanced Computing

# Collaborators and Customers

- Meshing Groups
  - Cubit (SNL) - Unstructured, Node Movement, Topology
  - NWGrid (PNNL) - Unstructured
  - Overture (LLNL) - Hybrid, Node Movement
- Applications
  - TAU3D (SLAC) - Unstructured, Node Movement, Top
  - Fischer (ANL) - Block Structured, Node Movement
  - Alegra (SNL) - Rezone
  - Sierra (SNL) - Adaptivity
- Research
  - Plassmann (Penn State)
  - Ollivier-Gooch (UBC)
  - Shashkov (LANL)

# Summary

The TSTT Center focuses on interoperable meshing and discretization strategies on complex geometries

- Short term impact through technology insertion into existing SciDAC applications
- Long term impact through the development of
  - a common mesh interface and interoperable and interchangeable mesh components
  - new technologies that facilitate the use of hybrid meshes
    - Discretization Library
    - Mesquite mesh quality improvement
- Working with SciDAC ISICs to ensure applicability of tools and interfaces

# Contact Information

TSTT

- Web Site: www.tstt-scidac.org
- David Brown: dlb@llnl.gov
- Lori Freitag Diachin: ladiach@sandia.gov
- Jim Glimm: glimm@ams.sunysb.edu

Mesquite

- Patrick Knupp: pknupp@sandia.gov

# Questions?